

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-328933

(43)Date of publication of application : 13.12.1996

(51)Int.Cl.

G06F 12/00

G06F 12/00

G06F 15/16

(21)Application number : 07-157063

(71)Applicant : NEC CORP

(22)Date of filing : 30.05.1995

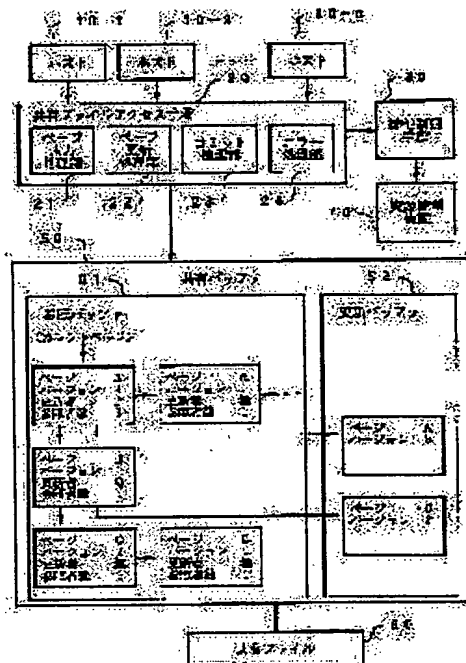
(72)Inventor : KARIYA YOSHIO

(54) FILE ACCESS CONTROL SYSTEM FOR PARALLEL PROCESSING SYSTEM

(57)Abstract:

PURPOSE: To provide a file access control system for parallel processing system by which the throughput of processing of data can be improved and the simultaneous executability of data processing can be improved by reducing overhead due to the exclusive control of file.

CONSTITUTION: In a parallel processing system for simultaneously performing access from respective hosts through a shared buffer 50 to a shared file 60, when updating a page, a shared file access means 20 sets information showing an updating person onto a reference buffer 51 and prepares the page of a new version on an update buffer 52 without updating any existent page on the reference buffer, so that page reference and update can be simultaneously executed. The information showing the number of reference persons is set onto the reference buffer 51 and used for opening the reference buffer. Thus, lock control for the unit of a page is unnecessary, reference and update can be processed in parallel, and waiting time due to the exclusive control is reduced.



LEGAL STATUS

[Date of request for examination] 30.05.1995

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2853608

[Date of registration] 20.11.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-328933

(43) 公開日 平成8年(1996)12月13日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 3 5	7623-5B	G 0 6 F 12/00	5 3 5 M
	5 1 4	7623-5B		5 1 4 A
15/16	3 4 0		15/16	3 4 0 V

審査請求 有 請求項の数 5 F D (全 7 頁)

(21) 出願番号 特願平7-157063

(22) 出願日 平成7年(1995)5月30日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 刈谷 善雄

東京都港区芝五丁目7番1号 日本電気株式会社内

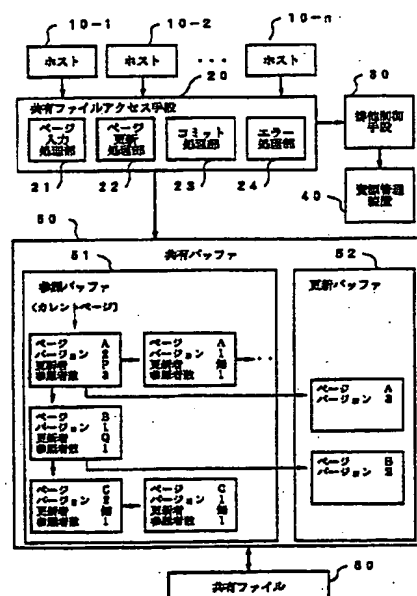
(74) 代理人 弁理士 松本 正夫

(54) 【発明の名称】 並列処理システムのファイルアクセス制御方式

(57) 【要約】

【目的】 ファイル排他制御によるオーバーヘッドを削減することにより、データの処理のスループットを向上し、データ処理の同時実行性を向上することができる並列処理システムのファイルアクセス制御方式を提供する。

【構成】 各ホストから共有バッファ50を介して共有ファイル60を同時アクセスする並列処理システムにおいて、共有ファイルアクセス手段20は、ページを更新する場合、参照バッファ51上に更新者を示す情報を設定し、参照バッファ上の既存のページを更新せずに、新しいバージョンのページを更新バッファ52上に作成することにより、ページ参照と更新の同時実行を可能とする。参照バッファ51上に参照者数の情報を設定し、参照バッファを開放するために使用する。これにより、ページ単位のロック制御を不要とし、参照と更新間の並行処理を可能とすると共に、排他制御による待ち時間を削減する。



【特許請求の範囲】

【請求項1】 各ホストから共有バッファを介し共有ファイルを同時アクセスする並列処理システムにおいて、前記共有バッファ上に、ページ参照用の参照バッファとページ更新用の更新バッファを設け、

前記参照バッファ上の各ページ毎に、該当ページのバージョン情報と更新者を示す情報と参照者数を示す情報を設定し、

ページを更新する場合、参照バッファ上の既存のページを更新せずに新しいバージョンのページを更新バッファ上に作成することにより、ページ参照とページ更新の同時実行を可能とする共有ファイルアクセス手段を備えることを特徴とする並列処理システムのファイルアクセス制御方式。

【請求項2】 前記バージョン情報は、ページ間で矛盾ないデータアクセスを行うために使用し、前記更新者情報は同一ページに対する同時更新を制約するために使用し、前記参照者数情報は前記参照バッファを解放するために使用し、

カレントページ以外の場合は前記参照者数が“0”になった時点で解放し、カレントページの場合は参照者数が“0”で、かつカレントページと同一で古いバージョンのページが存在しない場合のみ解放の対象とすることを特徴とする請求項1に記載の並列処理システムのファイルアクセス制御方式。

【請求項3】 前記共有ファイルアクセス手段は、ページの参照時において、必要ページを前記参照バッファから検索し、参照者が参照バッファを参照中ならばその参照バッファを参照し、当該参照者が更新中のときのみ更新バッファを参照し、必要ページが存在しない場合、前記共有バッファに参照バッファを割り当てて前記共有ファイルから必要ページを入力するページ入力処理部と、

ページの更新時において、当該更新者が当該ページを更新中の場合に更新バッファを更新可能な状態とし、他者が更新中の場合にエラーとし、更新者無しの場合に前記参照バッファから前記更新バッファに該当ページをコピーし、更新バッファを更新可能な状態とするページ更新処理部とを備えることを特徴とする請求項1又は2に記載の並列処理システムのファイルアクセス制御方式。

【請求項4】 前記共有ファイルアクセス手段は、コミット処理時において、自ら更新中の更新バッファを共有ファイルに出力し、参照バッファのカレントページとして付け替え、さらに参照バッファに対して、更新者無しの状態に設定し、参照者数から“1”を減じる処理を行なうコミット処理部と、

エラー処理時において、自ら更新中の更新バッファを廃棄して参照バッファに対して更新者無しの状態に設定すると共に、自ら更新中あるいは参照中である場合、参照バッファの参照者数から“1”を減じる処理を行なうエ

ラー処理部とを備えることを特徴とする請求項1、2又は3に記載の並列処理システムのファイルアクセス制御方式。

【請求項5】 前記ページ入力処理部は、参照バッファを割り当てて前記共有ファイルから入力したページのバージョンが、必要ページのバージョンと一致する場合、当該ページの参照者数に“1”を加え、バージョンが一致しない場合にエラーとし、

前記参照バッファに他者が参照中の必要ページと同一のページが存在する場合、当該ページのバージョンが一致する場合、当該ページの参照者数に“1”を加え、バージョンが一致しない場合、前記共有バッファに参照バッファを割り当てて前記共有ファイルから必要ページを入力することを特徴とする請求項3に記載の並列処理システムのファイルアクセス制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、各ホストから共有するファイルを備える並列処理システムに関し、特に、トランザクション処理の同時実行性を向上させるための並列処理システムのファイルアクセス制御方式に関するものである。

【0002】

【従来の技術】トランザクション処理の同時実行性を向上させるための排他制御方式に関する従来技術としては、例えば、特開平5-216742号公報（並列処理方法及び装置）、特開平2-144637号公報（データ構造に対するアクセス方法）、特開昭63-265354号公報（データ処理競合管理方法）等がある。

【0003】特開平5-216742号公報は、複数の処理ユニットで共有する大容量の計算用記憶領域として拡張領域を利用し、複数の処理ユニットによる大規模な配列の分散処理を可能とする方式を開示する。特開平2-144637号公報は、並列処理システムにおいて読取りをロックなしに行なうことにより、ロックに伴うオーバーヘッドを除くアクセス方法を開示する。特開昭63-265354号公報は、2つのファイルのレコードを同一形式で定義し、書き込みと読み込みの競合時に、一方を書き込み用、他方を読み込み用として選択して並列処理を行なう方式を開示する。これら従来における各方式では、何れも書き込み処理時に必ずロック制御を必要とする。

【0004】

【発明が解決しようとする課題】上述した従来方式を並列処理システムに適用する場合、並列処理システム全体の制御を行うことができるロック制御を必要とし、ロック制御のための大きなオーバーヘッドが生じ、かつ待ち状態発生によるデッドロックチェックのための並列システム全体の状態管理（ウェイトグラフ管理）を必要とするといった問題点があった。また、ファイルへの書き込みを含めその間中ロックすることが必要となり、トランザ

クション処理の同時実行性を阻害していた。

【0005】本発明の目的は、ファイル排他制御によるオーバーヘッドを削減することにより、データの処理のスループットを向上し、データ処理の同時実行性を向上することができる並列処理システムのファイルアクセス制御方式を提供することにある。

【0006】

【課題を解決するための手段】上記目的を達成するため、本発明は、各ホストから共有バッファを介し共有ファイルを同時アクセスする並列処理システムにおいて、前記共有バッファ上に、ページ参照用の参照バッファとページ更新用の更新バッファを設け、前記参照バッファ上の各ページ毎に、該当ページのバージョン情報と更新者を示す情報と参照者数を示す情報を設定し、ページを更新する場合、参照バッファ上の既存のページを更新せずに新しいバージョンのページを更新バッファ上に作成することにより、ページ参照とページ更新の同時実行を可能とする共有ファイルアクセス手段を備える構成としている。

【0007】他の態様では、前記バージョン情報は、ページ間で矛盾ないデータアクセスを行うために使用し、前記更新者情報は同一ページに対する同時更新を制約するために使用し、前記参照者数情報は前記参照バッファを解放するために使用し、カレントページ以外の場合は前記参照者数が“0”になった時点で解放し、カレントページの場合は参照者数が“0”で、かつカレントページと同一で古いバージョンのページが存在しない場合のみ解放の対象とする。

【0008】また、他の好ましい態様では、前記共有ファイルアクセス手段は、ページの参照時において、必要ページを前記参照バッファから検索し、参照者が参照バッファを参照中ならばその参照バッファを参照し、当該参照者が更新中のときのみ更新バッファを参照し、必要ページが存在しない場合、前記共有バッファに参照バッファを割り当てて前記共有ファイルから必要ページを入力するページ入力処理部と、ページの更新時において、当該更新者が当該ページを更新中の場合に更新バッファを更新可能な状態とし、他者が更新中の場合にエラーとし、更新者無しの場合に前記参照バッファから更新バッファに該当ページをコピーし、更新バッファを更新可能な状態とするページ更新処理部とを備える構成としている。

【0009】さらに、他の好ましい態様では、前記共有ファイルアクセス手段は、コミット処理時において、自ら更新中の更新バッファを共有ファイルに出力し、参照バッファのカレントページとして付け替え、さらに参照バッファに対して、更新者無しの状態に設定し、参照者数から“1”を減じるコミット処理部と、エラー処理時において、自ら更新中の更新バッファを廃棄して参照バッファに対して更新者無しの状態に設定すると共に、自

ら更新中あるいは参照中である場合、参照バッファの参照者数から“1”を減じるエラー処理部とを備える構成としている。

【0010】さらに、他の好ましい態様では、前記ページ入力処理部は、参照バッファを割り当てて前記共有ファイルから入力したページのバージョンが、必要ページのバージョンと一致する場合、当該ページの参照者数に“1”を加え、バージョンが一致しない場合にエラーとし、前記参照バッファに他者が参照中の必要ページと同一のページが存在する場合、当該ページのバージョンが一致する場合、当該ページの参照者数に“1”を加え、バージョンが一致しない場合、前記共有バッファに参照バッファを割り当てて前記共有ファイルから必要ページを入力する。

【0011】

【作用】本発明の並列処理システムのファイルアクセス制御方式では、共有バッファを有し、共有ファイルアクセス手段は、共有バッファ上で参照バッファと更新バッファを別々に備え、かつ共有バッファ上で排他制御を行うことにより、ページ単位のロック制御を不要としている。ページ単位のロック制御が不要であるため、待ち状態は発生せず、ウェイトグラフ管理も不要となり、ファイル入出力を含む間中の長時間ロックも発生しない。

【0012】

【実施例】本発明の好ましい実施例について図面を参照して詳細に説明する。図1において、本発明の一実施例による並列処理システムは、各ホスト10-1~10nと、共有ファイルアクセス手段20と、排他制御手段30と、資源管理装置40と、共有バッファ50と、共有ファイル60から構成される。

【0013】共有ファイルアクセス手段20は、ページ入力処理部21、ページ更新処理部22、コミット処理部23、エラー処理部24を備え、共有バッファ50を制御管理するための手段である。排他制御手段30は、共有ファイルアクセス手段20によって利用され、共有バッファ50に対する排他制御を行うための手段である。資源管理装置40は、排他制御手段30によって利用され、並列処理システム全体の排他制御を行うための装置である。

【0014】共有バッファ50は、参照バッファ51と、更新バッファ52から構成され、共有ファイルアクセス手段20によってバッファ管理される。参照バッファ51は、各バッファ毎の管理情報として、各ページ毎に該当ページのバージョンを示すバージョン情報B、更新トランザクションを示す更新者情報U、参照トランザクション数を示す参照者数情報Rを有する。

【0015】これらの情報は、共有ファイルアクセス手段20が共有バッファ50上で排他制御を行うために使用される。なお、バージョン情報Bとしては、一例としてタイムスタンプ情報が用いられ、各ページに対する更

新日時をセットし、各トランザクションの起動日時以降に更新されたページイメージをアクセスしないように制御することにより、ページ間の矛盾を発生させないようにする。

【0016】更新者情報Uは、同一ページに対する同時更新を制約し、あるいは更新者のみに更新中のバッファの参照を許可するために用いる。参照者数情報Rは、参照バッファ51の各バッファを解放するために用い、カレントページ以外の場合は参照者数の値が“0”になった時点で、LRU制御等によるバッファの解放対象とし、カレントページの場合は参照者数の値が“0”で、かつカレントページと同一で古いバージョンのページが存在しない場合に限りバッファの解放対象とする。共有ファイル60は、並列処理システムで共有するファイルである。

【0017】ここで、共有バッファ50の状態例について図1に示すページAによって説明する。図1において、ページAのコミット済の最新バージョンは、バージョン“2”であり、現時点で更新トランザクション“P”以外に2つの参照トランザクションの、計3つのトランザクションによりページAのバージョン2が同時に参照されている。さらに、参照バッファ51上に1世代前のバージョン“1”を保持し、他のページと同期を取って同一バージョンのデータ処理するために利用されている。更新バッファ52上のページAのバージョン“3”は、更新トランザクション“P”によって更新中である。

【0018】次いで、図2～図5は、共有ファイルアクセス手段20内のページ入力処理部21、ページ更処理部22、コミット処理部23、エラー処理部24における各フローチャートを示す。本発明の動作について、図1及び図2～図5を参照して説明する。まず、図1により動作の概要を説明する。

【0019】トランザクションが共有ファイル60の更新処理を行う場合、共有ファイルアクセス手段20は、更新するページのカレントページイメージを参照バッファ51に保存し、更新バッファ52に更新用バッファを確保することにより更新バッファ52上で更新を行なう。トランザクションが共有ファイル60の参照処理を行う場合には、カレントページイメージを参照し、参照と更新が同時実行できるようにする。

【0020】図2は、共有ファイルアクセス手段20のページ入力処理部21の動作である共有ファイル60内のデータ参照時のバッファ制御内容を示す。

【0021】データの参照時は、まず必要ページを参照バッファ51から検索する(ステップ201)。必要ページが存在しかつ当該参照者が更新中のときのみ(ステップ202)、更新バッファ52を参照する(ステップ208)。参照バッファが存在し、かつ参照者が参照バッファを参照済ならばそのバッファを参照する(ステッ

プ202)。

【0022】必要ページについて当該参照者が更新中でも参照済でもない場合(ステップ202)、同一ページが存在するかを検索する(ステップ203)。同一ページが存在する場合、同一バージョンのページかを判別し(ステップ204)、同一バージョンならば参照者数情報Rに“1”加えて参照する(ステップ207)。

【0023】同一バージョンのページが存在しない場合は(ステップ203、204)、参照バッファを割り当てて共有ファイル60からの入力を試みる(ステップ205)。必要とするバージョンより新しいバージョンしか入力できなかった場合は、当該参照トランザクションをエラーとして再処理させる(ステップ206)。また、必要とするバージョンを入力した場合には(ステップ206)、当該参照バッファの参照者数情報Rに“1”加えて参照する(ステップ207)。以上の処理により、トランザクション内の参照処理を矛盾なく実行する。

【0024】ただし、大量データを参照し集計するような場合、更新トランザクションが次々に到着している時には、バージョン不一致(ステップ206)となり集計処理を何度も繰り返すことにより、処理が終了しない事態が起こりうる。このような場合に対処するためには、集計処理の途中で何度かコミットを発行し集計処理を進展させるようにする必要がある。

【0025】図3は共有ファイルアクセス手段20のページ更新処理部22の動作である共有ファイル60内のデータ更新時のバッファ制御を示す。

【0026】データの更新時は、更新トランザクションが更新ページを未入力ならば、図2で説明したページ入力処理を行う(ステップ301)。当然、更新ページをバッファに保持している場合には、ステップ301の処理は不要となる。

【0027】当該更新者が既に当該ページを更新中の場合には、その更新バッファを更新中の状態とする(ステップ302)。当該更新者が更新中でない場合(ページ入力処理を行った場合を含む)、当該ページを他者が更新中であるならば、エラーとし再処理させる(ステップ303)。すなわち、ロールバックしてコミット時点から処理をやり直す。

【0028】参照バッファ51に当該ページが存在するが、他者が更新中でない場合には、参照バッファの更新者情報Uに当該更新トランザクションの情報を設定し(すなわち、自更新中の状態に設定する)、参照バッファから更新バッファにデータをコピーし、更新バッファを更新可能な状態とする(ステップ304)。言い換えれば、カレントページ(最新のページ)以外の更新はエラーとし、カレントページに対する更新については未更新である場合と他の更新者が既に更新済である場合に限り許可するように制御する。

【0029】従来方式の欠点である更新時のロック制御におけるオーバヘッドに関し、本実施例では、共有バッファ上で更新トランザクションを管理し、同一ページに対する更新トランザクションを1つのみに制約することによって、ロック制御を不要としている。

【0030】図4は共有ファイルアクセス手段20のコミット処理部23の動作であるコミット時のバッファ制御を示す。

【0031】コミット時は、自トランザクションの参照バッファを検索し（ステップ401）、参照バッファが存在しかつそのページをコミット発行者が更新中である場合には（402、403）、更新バッファを共有ファイル60に出力する（ステップ404）。さらに、当該更新バッファを参照バッファのカレントページとして付け替え（ステップ405）、参照バッファを更新済状態に変更する（ステップ406）。すなわち、参照バッファの更新者情報Uを更新者無しの状態にセットする。その後、参照者バッファの参照者数情報を“1”減らす（ステップ407）。

【0032】コミット発行者が参照バッファを参照中である場合（ステップ401、402、403）、参照者数情報から“1”を減らし（ステップ407）、参照者数が“0”となった参照バッファをLRU制御等による解放対象バッファとする。

【0033】図5は共有ファイルアクセス手段20のエラー処理部24の動作であるエラー発生時のバッファ制御を示す。

【0034】エラー発生時は、エラー発生者が更新中の更新バッファを廃棄し（ステップ501、502、503、504）、参照バッファの更新者を無しの状態に変更し（ステップ505）、参照トランザクション数（参照者数）を1減らす（ステップ506）。参照バッファに対しては、エラー発生者が参照中である場合（ステップ501、502、503）、参照者数から“1”を減らす（ステップ406）。

【0035】エラー発生時に、更新バッファを廃棄するだけで、ファイルのロールバックの必要がない理由は、ファイル上にはコミット済の最新データのみを保持し、共有バッファでのみ複数バージョンのデータを保持しているためである。

【0036】なお、共有バッファ50に対する同時アクセスのための排他制御は、必要があれば、排他制御手段30を使用して行う。排他制御手段30は、並列処理システム全体の排他処理を制御する資源管理装置40を使用して排他制御を行う。あるいは、共有バッファ上に基本同期命令であるテストアンドセット命令を持ち、排他

制御を行う方法やホスト間の通信手段を用いて排他制御を行う方法等が存在する。以上好ましい実施例をあげて本発明を説明したが、本発明は必ずしも上記実施例に限定されるものではない。

【0037】

【発明の効果】以上説明したように本発明によれば、並列処理システムにおいて、共有バッファのメモリ管理上で排他制御を行い、ページ単位のロック制御を不要とすることにより、参照と更新間の並行処理を可能とし、排他制御による待ち時間を削減することにより、データ処理のスループット及び並行処理度を向上させ、大量データ処理のシステムに対応させることが可能となる。

【0038】更新時のロック制御におけるオーバヘッドに関し、共有バッファ上で更新トランザクションを管理し、同一ページに対する更新トランザクションを1つのみに制約することによって、ロック制御を不要とすることができる。

【図面の簡単な説明】

【図1】 本発明の一実施例による並列処理システムの構成を示すブロック図である。

【図2】 本実施例の共有ファイルアクセス手段のページ入力処理部による処理内容を説明するフローチャートである。

【図3】 本実施例の共有ファイルアクセス手段のページ更新処理部による処理内容を説明するフローチャートである。

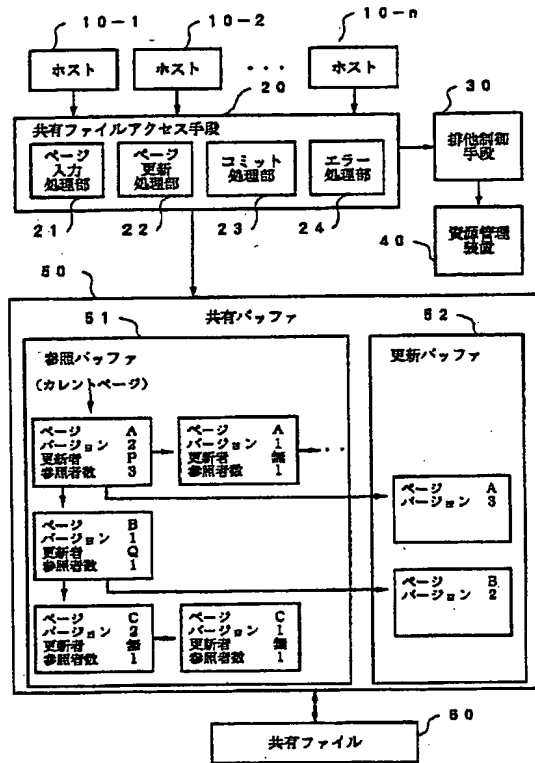
【図4】 本実施例の共有ファイルアクセス手段のコミット処理部による処理内容を説明するフローチャートである。

【図5】 本実施例の共有ファイルアクセス手段のエラー処理部による処理内容を説明するフローチャートである。

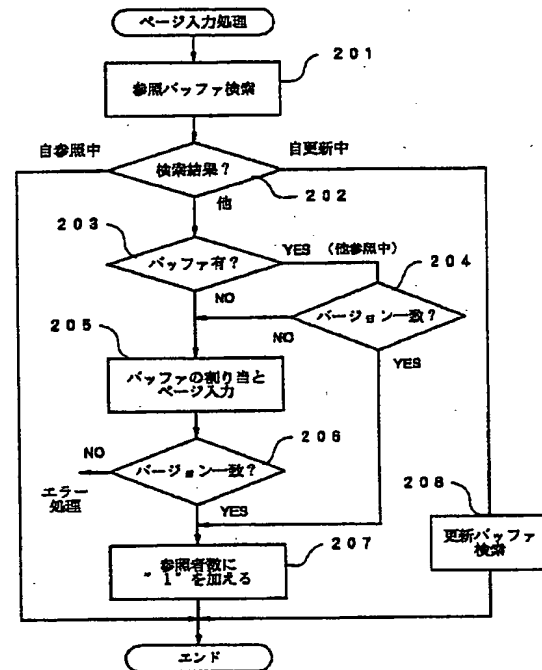
【符号の説明】

10-1～10-n ホスト
20 共有ファイルアクセス手段
21 ページ入力処理部
22 ページ更新処理部
23 コミット処理部
24 エラー処理部
30 排他制御手段
40 資源管理装置
50 共有バッファ
51 参照バッファ
52 更新バッファ
60 共有ファイル

【図1】

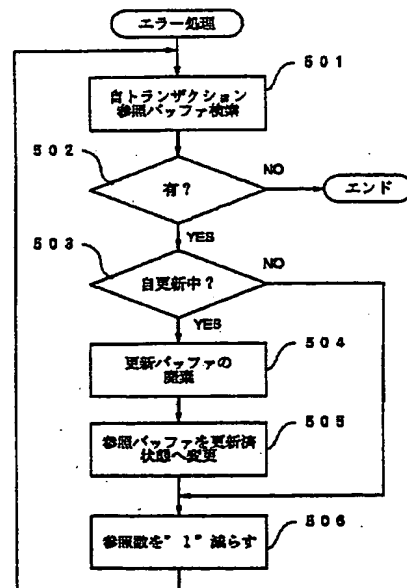
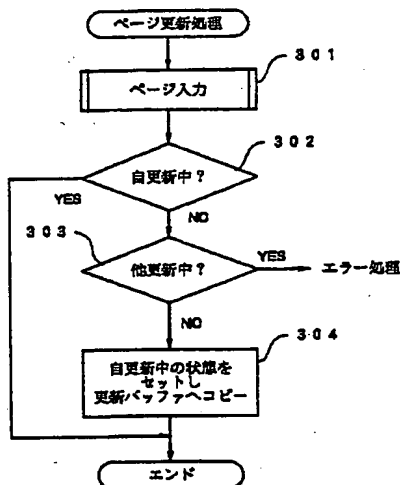


【図2】

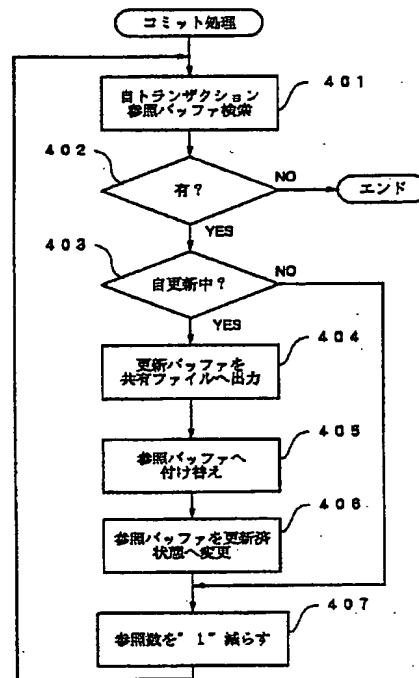


【図5】

【図3】



【図4】



Japanese Unexamined Patent Publication No. 4-337850

Published on November 25, 1992

Title of the Invention

DATABASE TRANSACTION AND QUERY PROCESSING SYSTEM

Abstract

[Object] To present a method and an apparatus, realized by utilizing an Intelligent Page Store, for providing concurrent and consistent access to a functionally separate transaction entity and a query entity to a shared database, while maintaining a single physical copy of most of the data.

[Construction] An Intelligent Page Store 10 contains shared nonvolatile storage, and an intelligent versioning mechanism 13 allows simultaneous access by the transaction entities 3, 9 and the query entities 15, 17 to shared data. The transaction entity is presented the current data and the query entity is presented a recent and consistent version of the data. A single copy of all but recently updated pages is maintained by the Intelligent Page Store. The query and transaction entities operate independently of each other and are separately optimized.